U.S. DEPARTMENT OF COMMERCE PATENT & TRADEMARK OFFICE

B/O Form PTO-1390 **Transmittal Letter to the United States Designated/Elected Office (DO/EO/US) Concerning a Filing Under *35 USC 371***		*Attorney's Docket Number* MERC3001/JEK
		U.S. Application Number (if known) 10/030106
International Application Number PCT/EP00/06833	*International Filing Date* 17 July 2000	*Priority Date Claimed* 19 July 1999
Title of Invention STACK OF OPERANDS AND METHOD FOR STACKING OF OPERANDS		
Applicant(s) for DO/EO/US Martin MERCK	Assignee	

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items under *35 USC 371*:

1. ☒ This is a **FIRST** submission of items concerning a filing under *35 USC 371*.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under *35 USC 371*.
3. ☒ This express request to begin national examination procedures *(35 USC 371(f))* at any time rather than delay examination until the expiration of the applicable time limit set in *35 USC 371(b)* and PCT Articles 22 and 39(1).
4. ☒ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. ☒ A copy of the International Application as filed *35 USC 371(c)(2)*.
   a. ☐ is transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☒ has been transmitted by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. ☒ A translation of the International Application into English *(35 USC 371(c)(2))*.
7. ☒ Amendments to the claims of the International Application under PCT Article 19 *(35 USC 371(c)(3))*
   a. ☐ are transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ have been transmitted by the International Bureau.
   c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
   d. ☒ have not been made and will not be made.
8. ☐ A translation of the amendments to the claims under PCT Article 19 *(35 USC 371(c)(3))*.
9. ☒ An oath or declaration of the inventor(s) *(35 USC 371(c)(4))*. (☐ Executed ☒ Unexecuted)
10. ☒ A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 *(35 USC 371(c)(5))*.

Items 11 to 16 below concern other document(s) or information included:

11. ☐ An Information Disclosure Statement under *37 CFR 1.97* and *1.98*.
12. ☐ An assignment document for recording. A separate cover sheet in compliance with *37 CFR 3.28* and *3.31* is included.
13. ☒ A **FIRST** preliminary amendment.
    ☐ A **SECOND** or **SUBSEQUENT** preliminary amendment.
14. ☐ A substitute specification.
15. ☐ A change of power of attorney and/or address letter.
16. ☒ Other items or information: 2 sheets formal drawings

Application Number (if known)	International Application Number	Attorney's Docket Number
10/030106	PCT/EP00/06833	MERC3001/JEK

	Calculations	PTO USE ONLY
17. The following fees are submitted: **Basic National Fee** *(37 CFR 1.492(a)(1)-(5))*: ☒ Search report has been prepared by the EPO or JPO . . . . . $890.00 ☐ International Preliminary Examination Fee paid to USPTO *(37 CFR 1.482)* . . . . . $710.00 ☐ No International Preliminary Examination Fee paid to USPTO *(37 CFR 1.482)* but International Search Fee paid to USPTO *(37 CFR 1.445(a)(2))* . . . . . $740.00 ☐ Neither International Preliminary Examination Fee *(37 CFR 1.482)* nor International Search Fee *(37 CFR 1.445(a)(2))* paid to USPTO . . . . . $1040.00 ☐ International Preliminary Examination Fee paid to USPTO *(37 CFR 1.482)* and all claims satisfied provisions of PCT Article 33(1)-(4) $100.00		
ENTER APPROPRIATE BASIC FEE AMOUNT	$ 890.00	
Surcharge of **$130.00** for furnishing the oath or declaration later than ☐ 20 ☐ 30 months from the earliest claimed priority date *(37 CFR 1.492(e))*.		
CLAIMS — **NUMBER FILED** — **NUMBER EXTRA** — **RATE**		
Total Claims: 11 -20 =	× $18.00	
Independent Claims: 2 -3 =	× $84.00	
Multiple Dependent Claims (if applicable)	+ $280.00	
TOTAL OF ABOVE CALCULATIONS	$ 890.00	
Reduction by ½ for filing by small entity, if applicable. Small Entity Status is asserted pursuant to 37 CFR 1.27 for this application.		
SUBTOTAL	$ 890.00	
Processing fee of **$130.00** for furnishing the English translation later than ☐ 20 ☐ 30 months from the earliest claimed priority date *(37 CFR 1.492(f))*.		
TOTAL NATIONAL FEE	$ 890.00	
Fee for recording the enclosed assignment *(37 CFR 1.21(h))*. The assignment must be accompanied by an appropriate cover sheet *(37 CFR 3.28, 3.31)*. **$40.00** per property.		
TOTAL FEES ENCLOSED	$ 890.00	
Amount to be:	Refunded:	
	Charged:	

a. ☒ A check in the amount of $890.00 to cover the fees is enclosed.

b. ☐ Please charge my **Deposit Account Number 02-0200** in the amount of $ _____ to cover the above fees. A duplicate copy of this sheet is enclosed.

c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to **Deposit Account Number 02-0200**. A duplicate copy of this sheet is enclosed.

Note: Where an appropriate time limit under *37 CFR 1.494* or *1.495* has not been met, a petition to revive *(37 CFR 1.137(a) or (b))* must be filed and granted to restore the application to pending status.

Customer 23364

BACON & THOMAS, PLLC
625 SLATERS LANE - FOURTH FLOOR
ALEXANDRIA, VIRGINIA 223124-1176
(703) 683-0500

DATE: 18 January 2002

Respectfully submitted,

J. Ernest Kenley
Attorney for Applicant
Registration Number: 19,179

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

International Patent Application No. PCT/EP00/06833

PCT/DO/EO/US

International Filing Date: 17 July 2000

Applicant: Martin MERCK

Atty Docket: MERC3001/JEK

For: STACK OF OPERANDS AND METHOD FOR STACKING OF OPERANDS

PRELIMINARY AMENDMENT

Commissioner for Patents
Washington, D.C. 20231

Sir:

This paper accompanies documents submitted to establish the U.S. national stage of the above-identified international patent application.

The international patent application was amended under PCT Article 34 and the claims as-amended are annexed to the International Preliminary Examination Report (IPER).

Before calculation of the filing fee and before examination, kindly amend the claims as annexed to the IPER as follows:

IN THE CLAIMS:

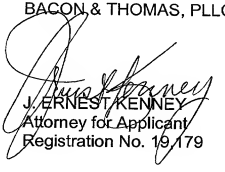
Please amend the claims as annexed to the IPER as shown on the appended APPENDIX OF CLAIMS, which includes amended and non-amended claims. Also appended hereto an APPENDIX OF MARKED UP CLAIMS showing the changes which have been made.

REMARKS

All rights are reserved to the original claimed subject matter. The claims have been amended to reduce the filing fees and to restate the inventive subject matter in clear terms. None of the amendments are intended to narrow any element of the

claims as they stood prior to amendment. Examination of the application as amended is respectfully requested.

Respectfully submitted,
BACON & THOMAS, PLLC



J. ERNEST KENNEY
Attorney for Applicant
Registration No. 19,179

Customer 23364

BACON & THOMAS, PLLC
625 Slaters Lane - 4th Floor
Alexandria, VA 22314-1176
Telephone: (703) 683-0500
Facsimile: (703) 683-1080

Date: January 17, 2002

S:\Producer\jek\MERCK - MERC3001\preliminary amendment.wpd

APPENDIX OF MARKED UP VERSION OF CLAIMS

4(Amended). An operand stack according to [any of claims 1 to 3] claim 1, characterized in that the operand stack is formed as a virtual stack for a virtual calculating machine.

5(Amended). An operand stack according to [any of claims 1 to 4] claim 1, characterized by an operand type checking device (S12-S14) which is activated at each read access to the operand memory (10, 32).

6(Amended). A calculating machine having an operand stack according to [any of claims 1 to 5] claim 1.

7(Amended). A smart card having an integrated virtual calculating machine according to [any of claims 1 to 6] claim 1.

11(Amended). A method according to [any of claims 8 to 10] claim 8, characterized in that a type check is performed at each read access to the operand memory (10, 32).

S:\Producer\jek\MERCK - MERC3001\appendix of marked up version of claims.wpd

APPENDIX OF CLAIMS

1. An operand stack for a calculating machine containing a processing unit processing individual operands according to a program, and the operand stack in which operands of different lengths are stored as a stack, characterized by a type memory (20, 31) with memory elements of constant length which stores for each operand stored in the operand memory (10, 32) its type information which contains information about the length of the relevant operand, the length of the particular operand type being stored in a table in dependence on the corresponding type code.

2. An operand stack according to claim 1, characterized in that the type memory (20) is formed as a stack with constant length stack elements separate from the operand memory.

3. An operand stack according to claim 1, characterized in that the type memory (31) is integrated operand by operand into the operand memory.

4(Amended). An operand stack according to claim 1, characterized in that the operand stack is formed as a virtual stack for a virtual calculating machine.

5(Amended). An operand stack according to claim 1, characterized by an operand type checking device (S12-S14) which is activated at each read access to the operand memory (10, 32).

6(Amended). A calculating machine having an operand stack according to claim 1.

7(Amended). A smart card having an integrated virtual calculating machine according to claim 1.

8. A method for operating an operand stack in a calculating machine wherein the stack elements of the operand stack are used for storing operands of different length, characterized in that a type memory element (20a, 20b; 31) of uniform length is created for each operand in the operand stack (10, 32), the type information stored in a type memory element contains length information about the length of the corresponding operand, and said length information is evaluated at each access to the operand memory, the length of the particular operand type being stored in a table in dependence on the corresponding type code.

9. A method according to claim 8, characterized in that the type memory elements are created in the form of a separate stack (20).

10. A method according to claim 8, characterized in that the type memory elements (31) are stored contiguously with the corresponding operand memory stack element (32).

11(Amended). A method according to claim 8, characterized in that a type check is performed at each read access to the operand memory (10, 32).

S:\Producer\jek\MERCK - MERC3001\appendix of claims.wpd

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Inventor:	Martin MERCK	Examiner:	Not Assigned
Serial No:	10/030,106	Art Unit:	Not Assigned
Filed:	18 January 2002	Atty Dkt:	MERC3001/JEK

For: STACK OF OPERANDS AND METHOD FOR STACKING OPERANDS

PRELIMINARY AMENDMENT

Commissioner for Patents
Washington, D.C. 20231

Sir:

Before examination, kindly amend the application as follows:

IN THE CLAIMS:

Please cancel claims 1-11 without prejudice or disclaimer.

Please add new claims 12-22 shown on the APPENDIX OF NEW CLAIMS.

REMARKS

All rights are reserved to the original claimed subject matter. The claims have been amended to reduce the filing fees and to restate the inventive subject matter in clear terms. None of the amendments are intended to narrow any element of the claims as they stood prior to amendment. Examination of the application as amended is respectfully requested.

The Commissioner is hereby authorized to charge any fees associated with this communication or credit any overpayment to **Deposit Account No. 02-0200**. A duplicate copy of this paper is attached.

Respectfully submitted,
BACON & THOMAS, PLLC

J. ERNEST KENNEY
Attorney for Applicant
Registration No. 19,179

625 Slaters Lane, 4th Floor
Alexandria, Virginia 22314
Telephone: (703) 683-0500
Facsimile: (703) 683-1080

Date: May 22, 2002

S:\Producer\jek\MERCK - 030106\preliminary amendment 2nd wpd

APPENDIX OF NEW CLAIMS

13 12. (New) An operand stack for a calculating machine containing a processing unit processing individual operands according to a program, and the operand stack in which operands of different lengths are stored as a stack, characterized by a type memory (20, 31) with memory elements of constant length which stores for each operand stored in the operand memory (10, 32) its type information which contains information about the length of the relevant operand, the length of the particular operand type being stored in a table in dependence on the corresponding type code.

14 13. (New) An operand stack according to claim 12, characterized in that the type memory (20) is formed as a stack with constant length stack elements separate from the operand memory.

15 14. (New) An operand stack according to claim 12, characterized in that the type memory (31) is integrated operand by operand into the operand memory.

16 15. (New) An operand stack according to claim 12, characterized in that the operand stack is formed as a virtual stack for a virtual calculating machine.

17 16. (New) An operand stack according to claim 12, characterized by an operand type checking device (S12-S14) which is activated at each read access to the operand memory (10, 32).

18 17. (New) A calculating machine having an operand stack according to claim 12.

19. 18. (New) A smart card having an integrated virtual calculating machine according to claim 12.

20 19. (New) A method for operating an operand stack in a calculating machine wherein the stack elements of the operand stack are used for storing operands of different length, characterized in that a type memory element (20a, 20b; 31) of uniform length is created for each operand in the operand stack (10, 32), the type information stored in a type memory element contains length information about the length of the corresponding operand, and said length information is evaluated at each access to the operand memory, the length of the particular operand type being stored in a table in dependence on the corresponding type code.

21 20. (New) A method according to claim 19, characterized in that the type memory elements are created in the form of a separate stack (20).

22 21. (New) A method according to claim 19, characterized in that the type memory elements (31) are stored contiguously with the corresponding operand memory stack element (32).

23 22. (New) A method according to claim 19, characterized in that a type check is performed at each read access to the operand memory (10, 32).

S:\Producer\jek\MERCK - 030106\appendix of new claims.wpd

Operand stack and method for operating an operand stack

In a calculating machine, be it a hardware machine or a virtual machine, operands are processed in a central processing unit. The processed operands are stored in an operand memory, the operands to be processed being read from the operand memory.

The operands are customarily stored using a memory stack. Such an operand memory stack (or more simply, operand stack) is organized in such a way that a memory location of given size is reserved for a certain number of stack elements of the operand stack, the stack elements of constant length being set up in this memory area. A stack pointer, formed for example as a counter, is incremented or decremented at each memory access.

To facilitate understanding of the invention, reference is made to Fig. 5 of the drawing which schematically shows prior art operand stack 100. Operand stack 100 contains stack elements 100*a*, 100*b*,... each of constant length. Said constant length, *Lmax*, is determined by the longest operand to be stored. In the shown example a reference value is stored in "bottom" stack element 100*a*. The second operand in stack element 100*b* is a byte value, as is the operand in third stack element 100*c*. An integer is located in the fourth stack element. A short value and a byte value are stored as operands in the fifth and sixth stack elements.

In the present example the longest occurring operands are "reference" in the first stack element and "integer" in the fourth stack element. Each stack element 100*a*, 100*b*,... occupies a memory location with length *Lmax*. Each stack element 100*a*, 100*b*,... with length *Lmax* occupies a memory location which is to include four addressable locations in the example considered here. When a further operand is placed on operand stack 100 the content of stack pointer 101 is incremented by "4" so that it points to the next free stack element. After an operand is read, the content of the stack pointer is decremented by "4."

The disadvantage of uniform size stack elements, that is, stack elements each with four smallest addressable locations in the present example, is the considerable waste of space in storing relatively short operands. In the present example only the operands in stack elements "1" and "4" are operands with maximum length *Lmax*, the

other operands in stack elements "2," "3" and "6" (byte values) being the shortest operands and occupying not even half of the available memory space with length *Lmax*. The short value at "5" occupies only half of the available space in the stack element.

When the operands stored in operand stack 100 are processed it should be ensured that the operands stored in the operand stack actually correspond to the operand type according to the program. However, a continuous type check is impossible with the organization of the operand stack outlined in Fig. 5. A check with the aid of a verification process is possible, but this involves a complete data flow analysis which means considerable effort.

The invention is based on the problem of providing an operand stack which optimizes the memory space requirement and further permits a continuous type check. Furthermore, there is to be provided a method for operating an operand memory which optimizes the memory requirement for the operand stack and allows a continuous type check.

To solve this problem the invention provides an operand stack having a type memory associated therewith, the type memory storing for every single stored operand the corresponding type information which contains length information about said operand. Said information prevents memory space from being wasted when operands of different length are stored; the information is instead stored extremely densely in the operand stack. The inventive operand stack has two basic organizational forms: in a first form the type memory is formed as a stack with constant length stack elements separate from the operand memory. In an alternative version the type memory is integrated into the operand memory, that is, each operand which can have one of a predetermined number of given lengths is directly contiguous to the corresponding type information.

Since the type information available for each operand contains length information about the operand, it is clear from the start how much memory space the particular operand requires. Upon a write operation to the operand memory, that is, when a new operand is placed on the operand stack, the type information is stored in connection with said operand. When the operand is read, the type information is then first

evaluated, and accordingly the stack pointer can be set so that the corresponding number of memory elements is read for the operand. The type information is binary-coded, for example as a four-digit code which is unique for each occurring operand type. From this code the corresponding length information can be gained with the aid of a table.

The type information stored in connection with each single operand permits a continuous check of operand type during processing of the operands. Before an operand is read from the operand stack, the type information is read in order to read the operand with the corresponding number of locations. The thus available type information can be compared with desired information of a checking program. If comparison is negative, that is, the pending operand type does not match the expected operand type according to the program, error handling is performed.

The invention thus achieves optimization of memory space, on the one hand, and provides information allowing a continuous check of operand type, on the other hand.

The inventive operand stack and inventive method for operating an operand stack can be employed in connection with a hardware calculating machine but also in connection with a virtual calculating machine. The abovementioned advantages are obtained in both cases.

In the following, some examples of the invention will be explained in more detail with reference to the drawing, in which:

Fig. 1 shows a schematic representation of an operand stack in connection with a type memory;

Fig. 2 shows a schematic flowchart illustrating the operation upon storage of an operand;

Fig. 3 shows a flowchart illustrating the operation when an operand is read from an operand stack, a type check of the operand being performed;

Fig. 4 shows a schematized representation of an embodiment of an operand stack alternative to the embodiment according to Fig. 1; and

Fig. 5 shows a schematic representation of a prior art operand stack.

As stated above in connection with Fig. 5, stack elements 100*a*, 100*b*,... each of constant length *Lmax* are provided in operand stack 100 in the prior art so that a

considerable amount of memory space is lost in the case of shorter operands due to the memory space not used. This lost amount of memory space is shown by hatching in Fig. 5.

Fig. 1 shows schematically a first embodiment of an operand stack according to the invention, likewise illustrating the method for organizing said operand stack.

As indicated by the representation on the left in Fig. 1, individual stack elements 10*a*, 10*b*,... of operand stack 10 are just as long as the operand to be stored therein. The length information for each operand is located in separate type memory 20, which is likewise organized as a stack. Stack elements 20*a*, 20*b* of type memory 20 all have constant length. Each stack element 20*a*, 20*b*,.... of type memory 20 stores a four-bit code which clearly identifies the type of associated operand. Thus, the type information for operand "1" in stack element 20*a* identifies a reference value. The type information also clearly defines the particular length of the operand. At the bottom of Fig. 1 corresponding stack pointer 12 is indicated for type memory 20. For read-write operation of type memory 20, stack pointer 12 is incremented or decremented. The value of stack pointer 12 corresponds to the address of the next free memory location.

Stack pointer 11 for operand memory 10 is not incremented or decremented with a constant value but in accordance with the length of the operand. Thus, a value corresponding to the length of the operand is added to stack pointer 11 when an operand is placed on operand stack 10, the content of stack pointer 11 being decremented by the length of the operand when said operand is read from operand memory 10. The value of stack pointer 11 corresponds to the address of the next free memory location. During operation of the calculating machine the height of operand stack 10 is continuously reduced and increased in accordance with the individual read and write operations.

The expert will see that operand stack 10 has variable length stack elements 10*a*, 10*b*,.... which are contiguous without a gap in the available memory space. Type memory 20 is created at another location in the memory.

Fig. 2 illustrates in the form of a flowchart a write operation placing an operand on operand stack 10. In step *S*1 the operand type to be stored is stored in type memory 20, in step *S*2 stack pointer 12 is incremented. In step *S*3 the operand is placed on

operand stack 10, and in step *S*4 stack pointer 11 of the operand stack is incremented in accordance with the type placed on operand stack 10, that is, in accordance with the length of the operand which is known from the type information. As described above, stack pointers 11 and 12 then point to the next free memory location.

Fig. 3 shows schematically the sequence in a read operation. In step *S*11 the top element of the stack is read from type memory 20. The value of stack pointer 12 is decremented by the length of a stack element, four bits in the example, since, as described above, stack pointer 12 points to the initial address of the next free stack element. The value of decremented stack pointer 12 thus forms the initial address for reading the top element in type memory 20. In the present example, the information read from type memory 20 is that the operand is a byte value.

In step *S*12 a check of the operand type is performed. This check is not the subject matter of the invention and will not be explained in any detail here.

In step *S*13 it is inquired whether the type is the expected operand type. If not, error handling is performed in step *S*14.

If the operand type corresponds to the expected type, the corresponding operand is read in step *S*15, that is, the operand with the length corresponding to the byte value is taken from the top stack element in operand stack 10 in Fig. 1. The value of stack pointer 11 is decremented by the length of the operand, i.e. the length of the byte value in the present example. The value of stack pointer 11 thus forms the initial address of the top element to be read of operand stack 10. The value of stack pointer 11 therefore corresponds to the address of the next free memory element again after reading.

Fig. 4 shows an organization of the operand stack alternative to Fig. 1. Operand stack 30 contains type memory elements 31 of constant length and operand memory elements 32 whose length depends on the particular type. Operand stack 30 in Fig. 4 can likewise be operated according to the sequence of Fig. 2 and Fig. 3, the newest operands being shown on the right in Fig. 4 while the newest operands are on the top of the stack in Fig. 1.

Claims

1. An operand stack for a calculating machine containing a processing unit processing individual operands according to a program, and the operand stack in which operands of different lengths are stored as a stack, characterized by a type memory (20, 31) with memory elements of constant length which stores for each operand stored in the operand memory (10, 32) its type information which contains information about the length of the relevant operand.

2. An operand stack according to claim 1, characterized in that the type memory (20) is formed as a stack with constant length stack elements separate from the operand memory.

3. An operand stack according to claim 1, characterized in that the type memory (31) is integrated operand by operand into the operand memory.

4. An operand stack according to any of claims 1 to 3, characterized in that the length of the particular operand type is stored in a table in dependence on the corresponding type code.

5. An operand stack according to any of claims 1 to 4, characterized in that the operand stack is formed as a virtual stack for a virtual calculating machine.

6. An operand stack according to any of claims 1 to 5, characterized by an operand type checking device (*S*12-*S*14) which is activated at each read access to the operand memory (10, 32).

7. A calculating machine having an operand stack according to any of claims 1 to 6.

8. A smart card having an integrated virtual calculating machine according to any of claims 1 to 7.

9. A method for operating an operand stack in a calculating machine wherein the stack elements of the operand stack are used for storing operands of different length, characterized in that a type memory element (20*a*, 20*b*; 31) of uniform length is created for each operand in the operand stack (10, 32), the type information stored in a type memory element contains length information about the length of the

corresponding operand, and said length information is evaluated at each access to the operand memory.

10. A method according to claim 9, characterized in that the type memory elements are created in the form of a separate stack (20).

11. A method according to claim 9, characterized in that the type memory elements (31) are stored contiguously with the corresponding operand memory stack element (32).

12. A method according to any of claims 9 to 11, characterized in that a type check is performed at each read access to the operand memory (10, 32).

Abstract

An operand stack (10) permits optimization of memory space and a continuous check of operand type by creating a type memory (20) which stores type information for each operand, said information comprising information about the length of the operand.

This length information available for each single operand permits the operands to be stored extremely densely, while the prior art uses uniform length stack elements for each operand, their length depending on the longest operand.

(Fig. 1)

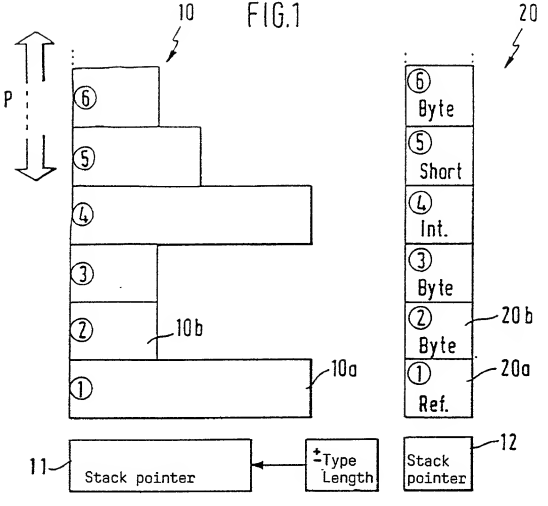


FIG.1

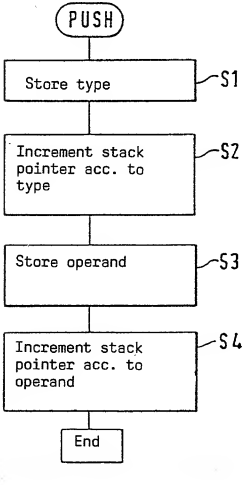


FIG.2

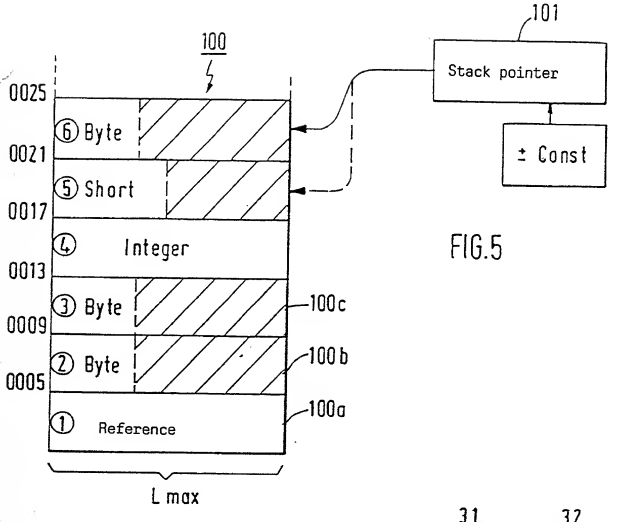


FIG.5

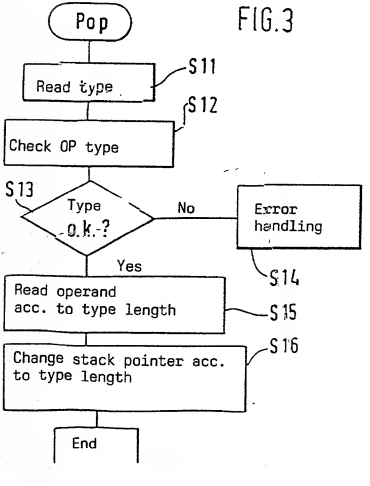


FIG.3

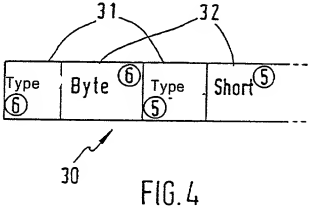


FIG.4

ATTORNEY/DOCKET NO: **MERC3001/JEK**

DECLARATION FOR PATENT APPLICATION AND APPOINTMENT OF ATTORNEY

As a below named inventor, I hereby declare that my residence, post office address and citizenship are as stated below next to my name; I believe that I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention (Design, if applicable) entitled:

STACK OF OPERANDS AND METHOD FOR STACKING OF OPERANDS

the specification of which (check one):

☐ is attached hereto, or ☒ was filed on: **17 July 2000** ✓ as U.S. Application Number or PCT International Application Number: **(PCT/EP00/06833) 10/030,106** and (if applicable) was amended on:

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment(s) referred to above. I acknowledge the duty to disclose information which is material to patentability as defined in *Title 37, Code of Federal Regulations, §1.56*. I hereby claim foreign priority benefits under *Title 35, United States Code §119* of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed.

PRIOR FOREIGN APPLICATION(S)			PRIORITY CLAIMED	
Number	Country	Day/Month/Year Filed	Yes	No
199 33 130.8	Germany ✓	19 July 1999 ✓	X	

☐ Additional Priority Application(s) Listed on Following Page(s)

I HEREBY CLAIM THE BENEFIT UNDER TITLE 35 U.S. CODE §119(E) OF ANY U.S. PROVISIONAL APPLICATIONS LISTED BELOW.	
Application Number	**Day/Month/Year Filed**

☐ Additional Provisional Application(s) Listed on Following Page(s)

I hereby claim the benefit under *Title 35, United States Code, §120* of any United States application(s) or PCT international application(s) designating The United States of America listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in that/those prior application(s) in the manner provided by the first paragraph of *Title 35, United States Code, §112*, I acknowledge the duty to disclose information which is material to patentability as defined in *Title 37, Code of Federal Regulations, §1.56* which became available between the filing date of the prior application(s) and the national or PCT international filing date of this application:

Application Number	Filing Date	Status - Patented, Pending or Abandoned

☐ Additional US/PCT Priority Application(s) listed on Following Page(s)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under *section 1001 of title 18 of the United States Code* and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: I (We) hereby appoint as my (our) attorneys, with full powers of substitution and revocation, to prosecute this application and transact all business in the Patent and Trademark Office connected therewith: J. Ernest Kenney, Reg. No. 19,179; Eugene Mar, Reg. No. 25,893; Richard B. Fichter, Reg. No. 26,382; Thomas J. Moore, Reg. No. 28,974; Joseph DeBenedictis, Reg. No. 28,502; Benjamin E. Urcia, Reg. No. 33,805; and

I(we) authorize my(our) attorneys to accept and follow instructions from **Klunker Schmitt-Nilson Hirsch** regarding any matter related to the preparation, examination, grant and maintenance of this application, any continuation, continuation-in-part or divisional based thereon, and any patent resulting therefrom, until I(we) or my(our) assigns withdraw this authorization in writing.

Send correspondence to: Customer 23364

BACON & THOMAS, PLLC
625 Slaters Lane - 4th Floor
Alexandria, VA 22314-1176

Telephone Calls to: **J. Ernest Kenney**
(703) 683-0500

FULL NAME OF FIRST OR SOLE INVENTOR	CITIZENSHIP
Martin MERCK	Germany ✓
RESIDENCE ADDRESS Toräckerweg 12 97084 Würzburg, Germany DEX	POST OFFICE ADDRESS IS THE SAME AS RESIDENCE ADDRESS UNLESS OTHERWISE SHOWN BELOW
DATE 29. April. 2002	SIGNATURE

☒ *See following page(s) for additional joint inventors.*